



# Open Systems Versus New Technology: A COTS Conundrum

**Presented To: National Defense Industrial Association  
5<sup>th</sup> Annual Systems Engineering Conference  
21-24 October 2002; Tampa, FL**

**Presented By: Walter J. (Jim) Hall, Jr.  
Senior Staff Engineer  
Smiths Aerospace Electronic Systems – Germantown  
20501 Goldenrod Lane  
Germantown, MD 20876 USA  
Ph: +1 301 428 6683 Fax: +1 301 428 6478  
[Jim.hall@smiths-aerospace.com](mailto:Jim.hall@smiths-aerospace.com)**

# Overview

# Defining and Open System

The Tri-Service Open Systems Architecture Working Group defines an Open System as:

"...a system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered components to be utilized across a wide range of systems with minimal changes, to interoperate with other components on local and remote systems, and to interact with users in a style that facilitates portability. An OS is characterized by the following:

- "Well defined, widely used, non-proprietary interfaces/protocols, and
- "Use of standards which are developed/adopted by industrially recognized standards bodies, and
- "Definition of all aspects of system interfaces to facilitate new or additional systems capabilities for a wide range of applications, and
- "Explicit provision for expansion or upgrading through the incorporation of additional or higher performance elements with minimal impact on the system."

# Smiths Open System Definition

An architecture that allows insertion or extraction of either hardware or software functions with a minimal of design and integration risk and cost. This definition requires:

- 1) A backplane/bus structure that is supported as an industry standard.
- 2) A common form factor for hardware elements (boards).
- 3) A common and simple Application Programmer's Interface for software.
- 4) A common and commercially available software and integration toolset.

# Smiths Approach to Realizing an Open System

- Industry Standard Architecture and Form Factor
  - VME64 (6U & 3U)
  - cPCI (6U & 3U)
  - PC/104 Plus
- Industry Standard Operating System
  - VxWorks
- COTS Tools for Software, Integration, and Test
  - WindRiver Tornado

***Implied, and critical is that hardware elements and software language used must be supported by industry and tools***



# The Conundrum

- So what's the problem

Most military equipment has requirements that cannot be met within these constraints

- Environment

- Functions

- Real-Time

- Unique Interfaces

- State-of-the-Art Applications (leading COTS)

- Physical

- Size, Weight, Power

- So we have a conundrum

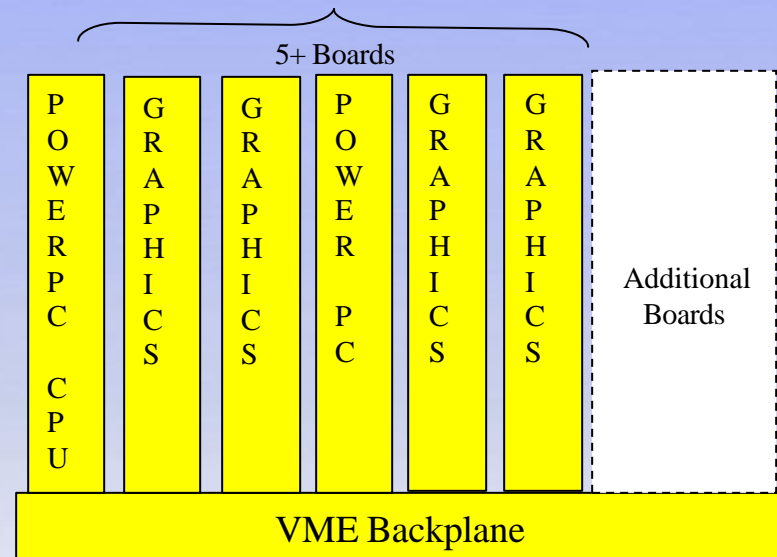
“a paradoxical, insoluble, or difficult problem; a dilemma.”

# How Do You Solve the Conundrum

- You don't! But, you *almost* can.
- Smiths Design Approach
  - Identify the elements that are inconsistent with an Open System and encapsulate them
  - Develop everything else as an Open System
    - Standard Architecture and Form Factor
    - COTS/ROTS Boards
    - VxWorks Real-Time Operating System
    - COTS Tools
  - Embed the unique function in the Open System
    - Minimize the uniqueness as much as possible
    - Minimize the impact of the unique elements

# Example - Display Processing

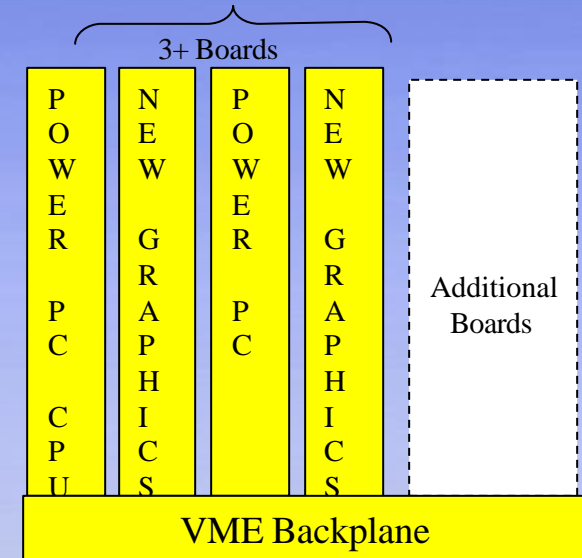
- Requirements
  - Drive 4 Independent Displays
  - Minimize cost, size, weight, power
- Conundrum -
  - Required 5+ COTS Boards
    - 1 Display Channel per Board
    - Requires General Processor
  - OSA Implementation
    - Too large
    - Too expensive





# Display Processing Solution 1

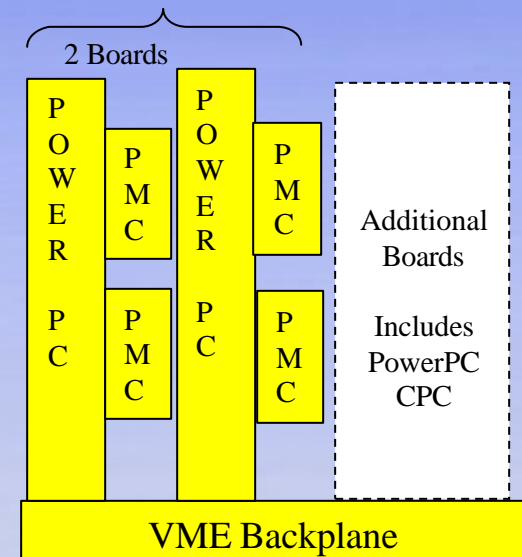
- Need to Reduce Size and Cost
  - Reduce the Number of Boards
    - Develop New Graphics Board
    - Team with VME Board Developer
    - Two Graphics Channels per Board
  - Eliminated 2 Graphics Boards
- Open System Effect
  - Graphics Board Fully VME Compliant
  - VxWorks Driver Provided with Board
- Fully Open System Solution



**COTS/ROTS OSA  
With Specially Developed Board**

# Display Solution 2

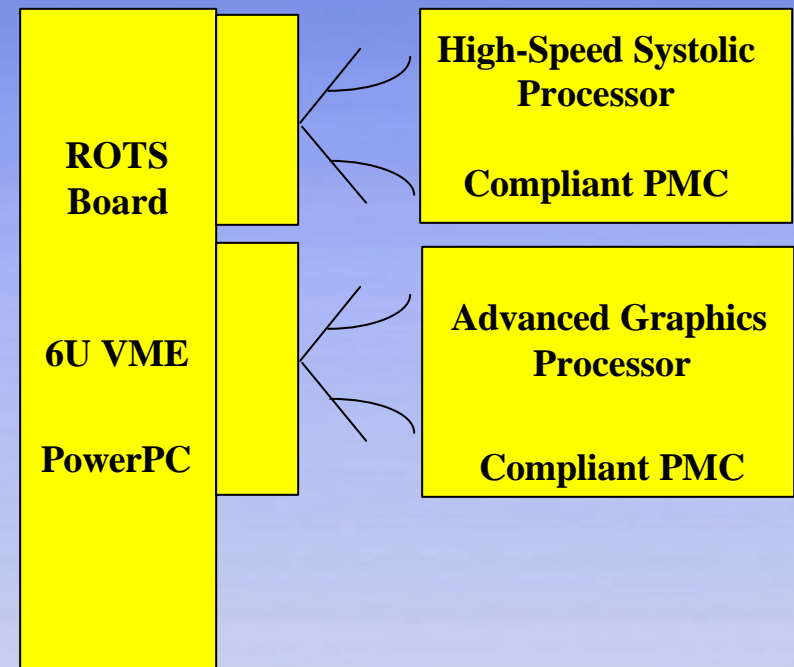
- Same Requirement - 4 Independent Displays
- Need Further Reduction in Size & Weight
  - Reduce the Board Count
  - Eliminate Use of CPU
- Solution - 2 Displays per Standalone Board
  - Develop PMC That Drives a Display
  - (PMC can drive 2 displays)*
  - Combine PMC & COTS PowerPC Board
  - Two Board Solution - Standalone
- Open System Effect
  - Uses COTS Power PC
  - Standard PMC
  - OpenGL Graphics I/F for Software
  - Unique Driver I/F for Graphics Device



Modified COTS/ROTS OSA  
With Unique PMCs and Drivers

# Example - Image Processing

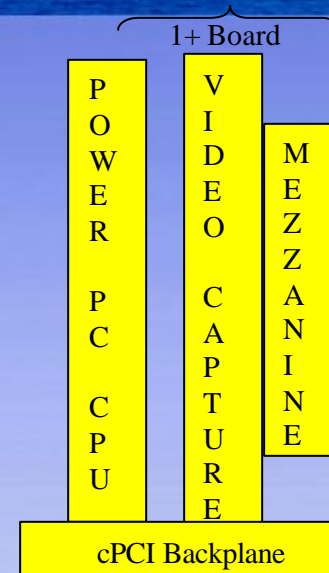
- The Requirement
  - Correlate Sensor Image with Geo-Registered Image
    - Perform Warp/Morph, Rotation, & Correlation
    - Extract Coordinates of Selected Object(s)
  - In Real-Time in a Cockpit
- The Conundrum
  - Currently Done On the Ground
  - Non-Real-Time Process (very long)
- The Solution - Single Board
  - High-Speed Systolic Processor (HSSP)
  - Embed HSSP on Standard PMC
  - Put HSSP and Graphics PMCs on COTS PowerPC
- Open System Effect
  - Uses COTS Power PC
  - Standard PMCs
  - OpenGL Graphics I/F for Software
  - Unique Driver I/F for Graphics Device and HSSP



*Single VME Board Implementation  
For Real-Time Image Correlation  
In Military Open System*

# Example - Video Recording and Compression

- The Requirement
  - Record 4 Video Channels
  - Record 2 Audio Channels
  - Reduce Storage (Compress)
  - Playback 1 Channel (while recording)
  - In Real-Time in a Military Vehicle
- The Conundrum
  - COTS Solutions Too Large
- The Solution - Single 3U cPCI Board+
  - Develop a Unique Baseboard
  - Develop a Unique Mezzanine Board
- Open System Effect
  - Standard cPCI Architecture and Form Factor
  - MPEG 2 Standard Compression
  - Unique Board and Mezzanine
  - Unique Drivers for Hardware I/Fs



## Two Board Implementation

### CPU (PowerPC Processor)

Performs CPU Functions

Manages Video Board

### Video Capture Board & Mezzanine

4 Video Input, 1 Output

2 Audio Input & Record

Video Capture/Framing

Video Compression

Video Playback (1 channel)



# Software Architecture

- The Requirement  
*Develop an Open System (Industry Standard Real-Time Operating System, COTS Tools, and Standard API)*
- The Conundrum  
*Implementing Elements for Which There is No Standard*
- The Solution
  - Use Standard Architecture and COTS/ROTS
  - Use VxWorks Real-Time Operating System
  - Use Tornado Development Tools
  - Use OpenGL and Other Standards
  - Develop and Integrate Unique I/F Drivers Only As Required
- Open System Effect
  - **Mostly** Open System
  - Minimized Uniqueness
- **OR - Develop a Standard API to Driver Uniqueness from Application**



# Summary

- Open Systems Are Highly Desirable
  - They allow insertion or extraction of either hardware or software functions with a minimal of design and integration risk and cost
- Typical Approach to Achieving an Open System
  - Use an Industry Standard Architecture (VME, cPCI, PC/104-Plus)
  - Use an Industry Standard Operating System (VxWorks)
  - Use Industry Standard Tools (Tornado)
  - Use COTS/ROTS Boards and Devices
- Unique Military Requirements Create A Conundrum
  - New Functionality
  - Unique Interfaces
  - Harsh Environment
  - Size, Weight, Power, and Cost Constraints
- A Solution for Military Equipment
  - Use an Open System 'Core'
  - Minimize the Unique Elements in the System
  - Mitigate the Uniqueness

***Do the Best You Can, You Can Get Very Close to an Open System***